

академия
больших
данных

mail.ru
group

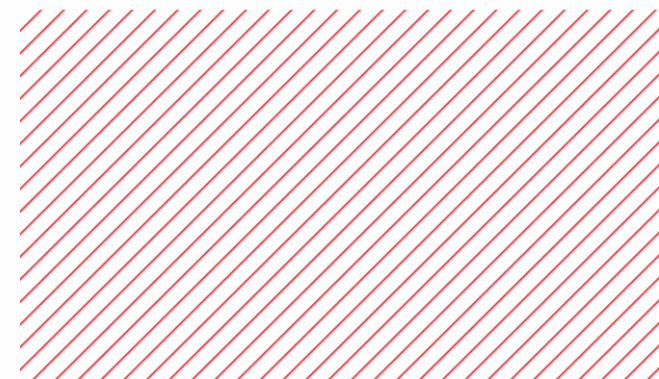


Обработка изображений на мобильных устройствах

Лекция 8

Распознавание текста

Евгений Мясников





Структура презентации

Распознавание текста. Источники данных и приложения.

История развития методов распознавания текста. Сопоставление шаблонов и структурный анализ. Метод моментов. Фурье-дескрипторы. Графовые модели. Синтаксический и алгебраический подход.

Нейросетевой подход. Основные этапы распознавания текста на цифровом изображении: детектирование текста, сегментация и распознавание символов.

Современные модели и библиотеки. Особенности реализации для мобильных устройств.

Примеры распознавания текста в Android Studio.



Распознавание текста. Источники данных.

OCR – Оптическое распознавание символов
(англ. Optical character recognition)

Цель – извлечение текстовой информации из изображений



Распознавание текста. Источники данных.

Сканирующие устройства:

- Планшетные
- Ручные
- Протяжные
- Проекционные
- Слайд-сканеры

Видео- и Фотокамеры

Источники в цифровом виде



Распознавание текста. Приложения.

Оцифровка существующей печатной продукции (книги, газеты, журналы и т.д.)

Считывание и распознавание идентификационных документов, особенно с машиночитаемыми строками (аэропорт, таможня, ...)

Распознавание автомобильных государственных регистрационных знаков

Ввод информации с банковских чеков (номера счетов, чеков, суммы)

Векторизация карт, планов и чертежей (номера домов, отметки высот и т.п.)

Ввод информации о договорах (контактные данные контрагентов)



Распознавание текста. Приложения.

Использование слепыми:

Распознавание с последующим воспроизведением аудио или симуляцией шрифтом Брайля.

В системах автономной навигации:

Чтение дорожных указателей

При факсимильной передаче:

Передача текста вместо графических данных

Извлечение данных из изображений (рекомендательные системы, безопасность и т.п.)

История развития методов распознавания текста

История развития методов распознавания текста

1929 / 1933 – первые патенты по OCR в Германии и США (Tausheck, Handel)

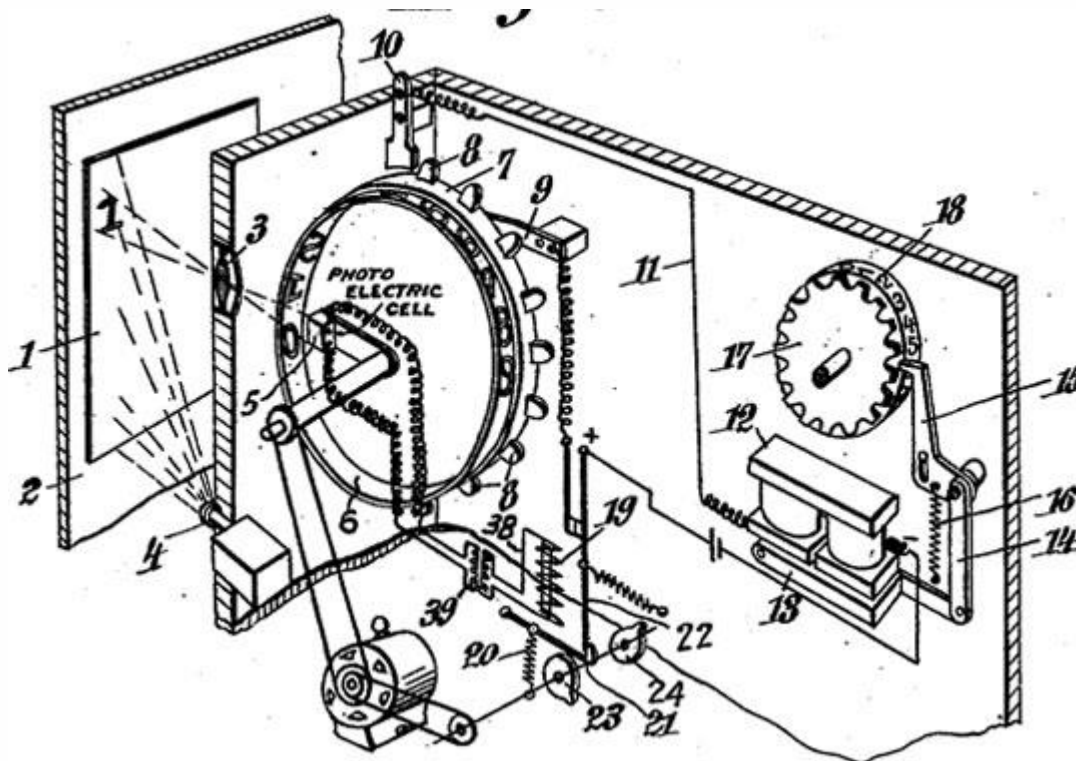
Электромеханическое оптическое устройство.

Принцип: сопоставление шаблонов

Изображение с текстом размещалось перед читающей машиной.

С внутренней стороны объектива вращалось колесо с отверстиями в виде букв.

При совпадении изображения и отверстия по форме, печатный барабан поворачивался на нужную букву и печатал ее на бумаге.



Патентный чертёж машины Таушека

G. Tausheck Reading Machine, 1929

История развития методов распознавания текста

Сопоставление шаблонов

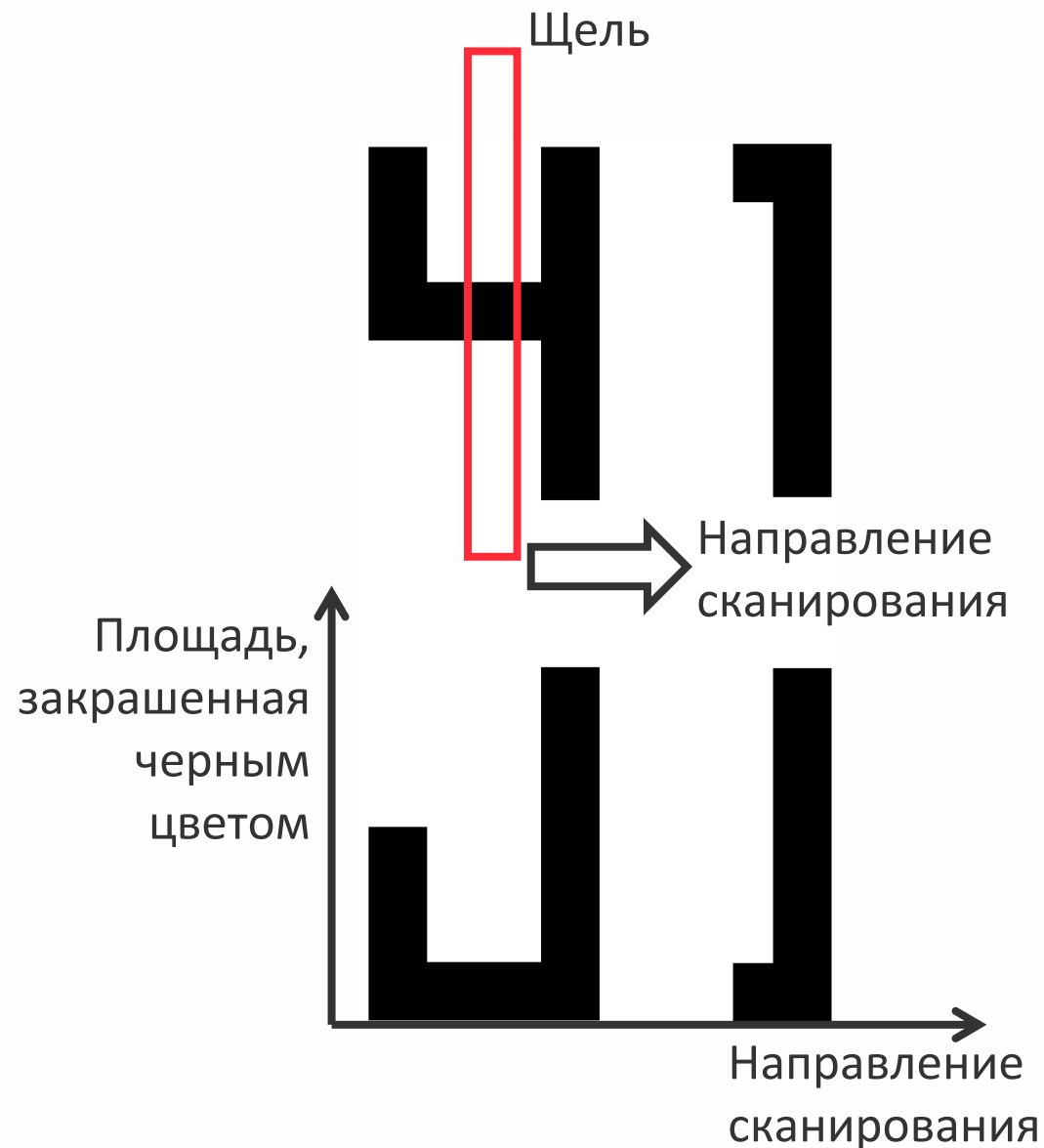
1956 Kelner and Glauberman

Предварительно совмещенный символ сканируется с использованием щели, через которую отраженный бумаги свет попадает на фотодетектор. Значение квантуется и оцифровывается.

Достигается инвариантность в направлении сканирования. При достаточно длинной щели достигается некоторая инвариантность к вертикальному положению цифры.

Символы как правило представляются связными компонентами и при достаточном расстоянии могут быть разделены друг от друга автоматически

Достигается снижение размерности



История развития методов распознавания текста

Сопоставление шаблонов

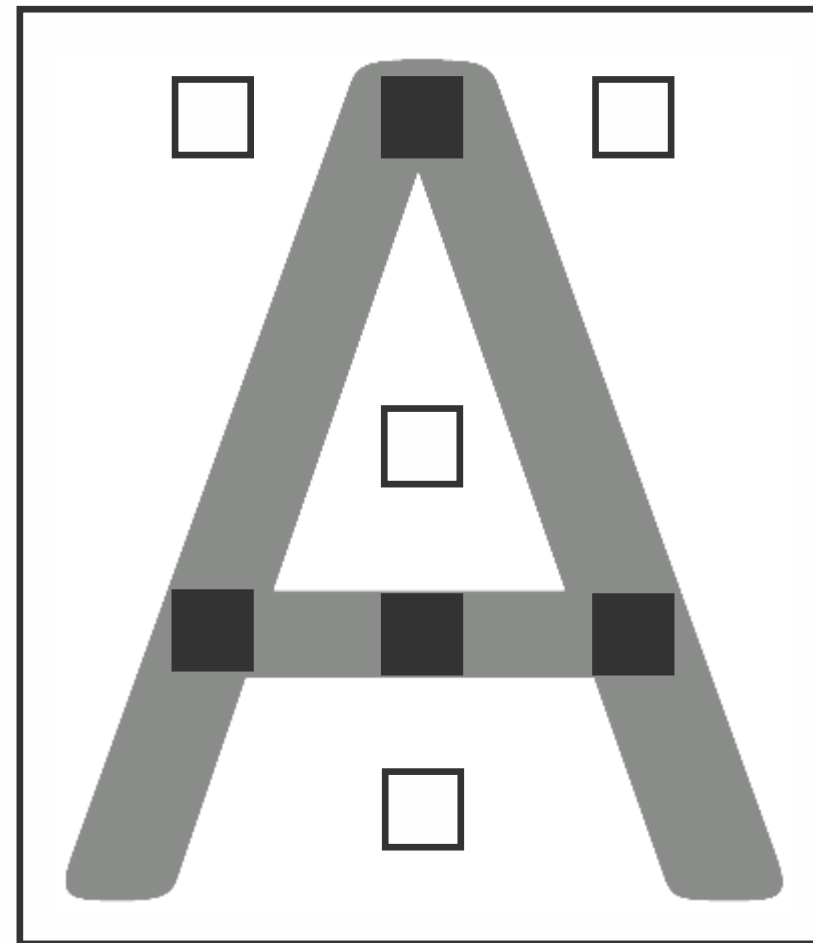
1956 peephole method

Символ выравнивается и используется сопоставление с использованием логической схемы по черным или белым значениям, наблюдаемым через «глазки», расположенные в соответствии с некоторой схемой

Пример:

1957 – система ERA (Electric Reading Automation) от Solatron Electronics Group Ltd.

Использовалось 100 «глазков», скорость чтения - 120 символов /сек.



История развития методов распознавания текста

Структурный анализ

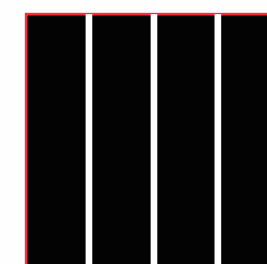
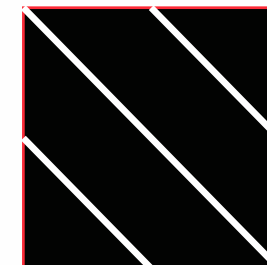
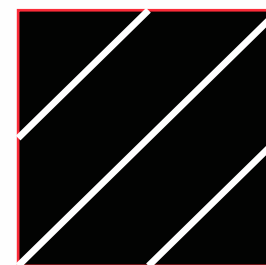
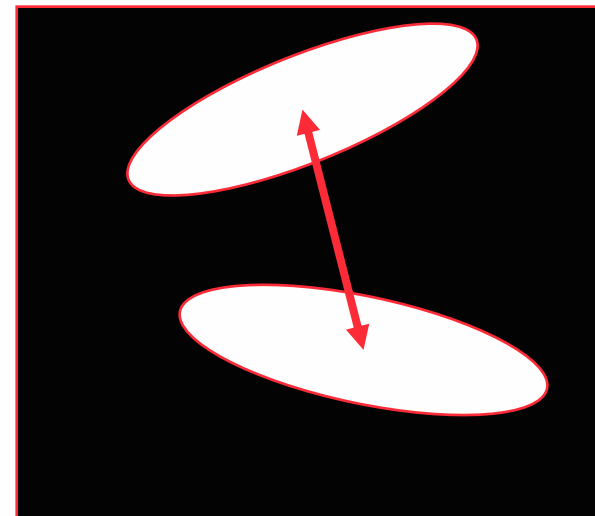
Структурный анализ – надежда на распознавание рукописных символов

Идея: структура целого может быть разбита на части, описана признаками частей и отношениями между ними

Вопрос: как выбрать признаки и отношения

Пример: Slit / Stroke анализ

- подсчет количества наблюдаемых в «вырезах» черных областей (1954, Rohland)
- подсчет переходов через черное для равномерно распределенных по изображению линий вертикальных, горизонтальных и диагональных (1961, Weeks)



История развития методов распознавания текста

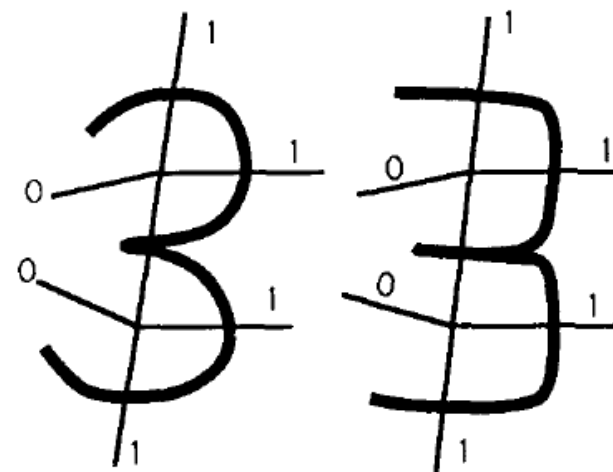
Структурный анализ

Использование «зондов» - специальным образом расположенных линий

Использование разнообразных тестов:

- максимальное количество пересечений вертикальной прямой
- минимальное расстояние до прямой

Недостаток: невозможно различить при большом количестве классов



История развития методов распознавания текста

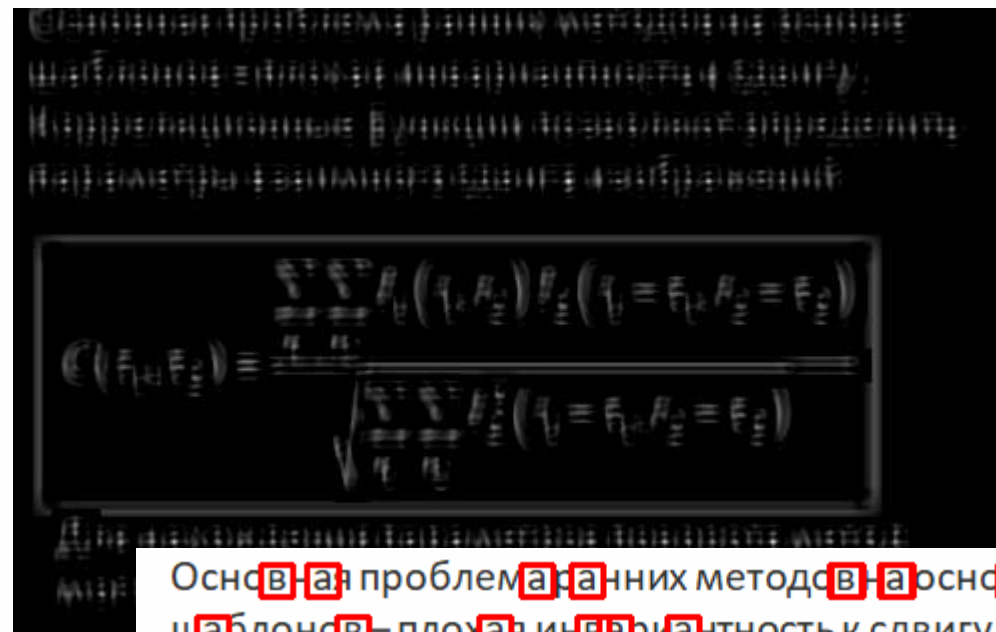
Корреляция

1950-е гг.

Основная проблема ранних методов на основе шаблонов – плохая инвариантность к сдвигу. Корреляционные функции позволяют определить параметры взаимного сдвига изображений

$$C(\tau_1, \tau_2) = \frac{\sum_{t_1} \sum_{t_2} I_1(t_1, t_2) I_2(t_1 - \tau_1, t_2 - \tau_2)}{\sqrt{\sum_{t_1} \sum_{t_2} I_2^2(t_1 - \tau_1, t_2 - \tau_2)}}$$

Для нахождения параметров поворота метод мог быть использован со множеством шаблонов



Основная проблема ранних методов на основе шаблонов – плохая инвариантность к сдвигу. Корреляционные функции позволяют определить параметры взаимного сдвига изображений

$$C(\tau_1, \tau_2) = \frac{\sum_{t_1} \sum_{t_2} I_1(t_1, t_2) I_2(t_1 - \tau_1, t_2 - \tau_2)}{\sqrt{\sum_{t_1} \sum_{t_2} I_2^2(t_1 - \tau_1, t_2 - \tau_2)}}$$

Для нахождения параметров поворота метод мог быть использован со множеством шаблонов

История развития методов распознавания текста

Метод моментов

Моменты:

$$m_{\alpha\beta} = \sum_{t_1} \sum_{t_2} t_1^\alpha t_2^\beta I(t_1, t_2)$$

Моментные инварианты:
момент инерции, эксцентриситет, ...

$$\Phi_1 = m_{20} + m_{02}, \quad \Phi_2 = (m_{20} + m_{02})^2 + 4m_{11}^2$$

Первые реализации:

1962 - Alt из National Bureau of Standards (NBS)
Моменты до 3 порядка использовались для нормализации, до 12 порядка – для классификации

1987 Cash and Hatamian
центральные моменты до 3 порядка - для классификации. 95 % точности, 62 класса, буквы, цифры

История развития методов распознавания текста

Ряды Фурье.

1960 гг. Cogriff, Zahn and Roskie's.

1 способ представления:

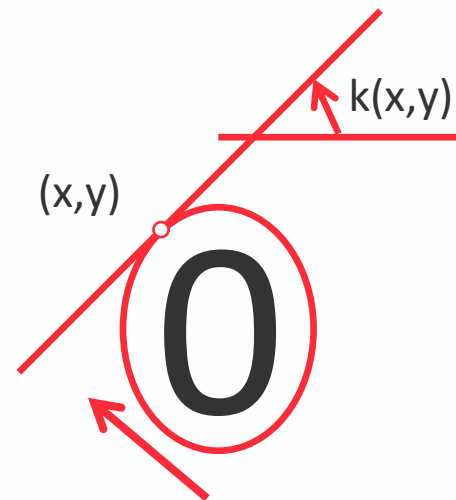
основанный на вычислении угла наклона касательной:

$$\kappa(s) = \kappa(x_1(s), x_2(s))$$

Ряд коэффициентов:

$$c_k = \frac{1}{L} \int_0^L k(s) \exp\left(-i \frac{2\pi}{L} sk\right) ds$$

Фурье-дескрипторы (амплитуды, фазы) инвариантны к преобразованиям подобия





История развития методов распознавания текста

Ряды Фурье. Способ 2

Granlund, Persoon and Fu


2 способ представления:
комплексная функция (CF)

Движущаяся вдоль границы точка порождает комплексную функцию
 $u(s) = x(s) + i y(s)$

Вычисляется ряд:

$$c_k = \frac{1}{L} \int_0^L u(s) \exp\left(-i \frac{2\pi}{L} sk\right) ds$$

Проведенные тогда исследования показали уровень ошибок ~10%,
14 сек/символ на машине того времени



История развития методов распознавания текста

Графовые модели.

Штрихи и отношения между ними представляются в виде графа. Для сопоставления описаний используется изоморфизм

Пример: 1960, Sherman

Граф строился по рукописной цифре так, что все узлы степени 2 игнорировались. Имели значения окончания линий, пересечения,...

Представления – матрица смежности

2 4 6 8



История развития методов распознавания текста

Синтаксический и алгебраический подход.

Сканируя размеченную плоскость в определенном порядке строим цепочку меток.

Задача состоит в конструировании автомата, который принимает строку как принадлежащую классу.

Должен быть сконструирован язык, грамматика которого принимает строку.

Помимо образов, с примитивами связаны атрибуты

Пример:

$G: LC \rightarrow L \mid C \mid L \otimes C$

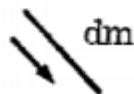
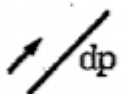
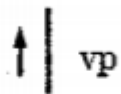
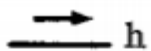
$L \rightarrow l$ - линия

$C \rightarrow c$ — окружность

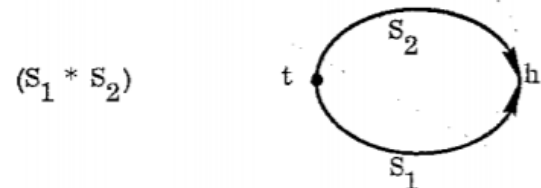
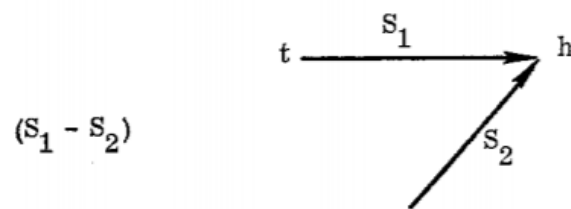
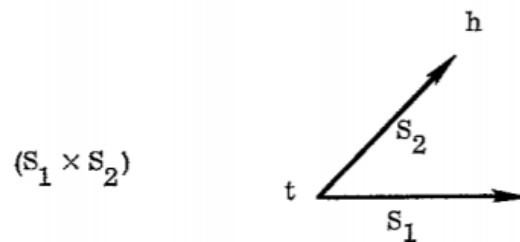
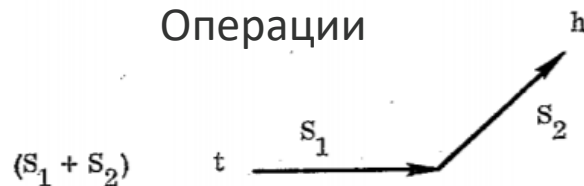
История развития методов распознавания текста

Синтаксический и алгебраический подход.

Примитивы



Операции

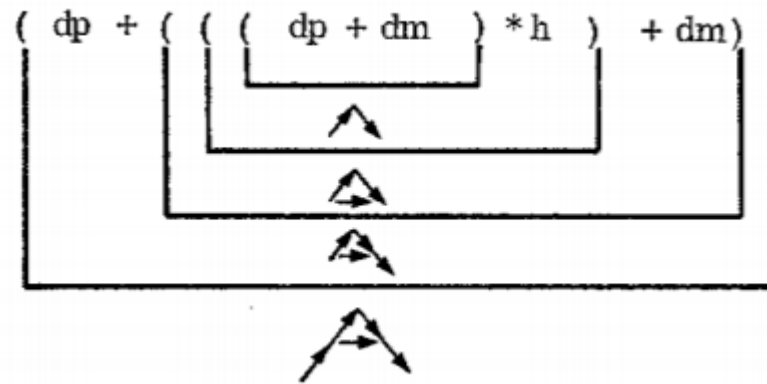


Описания на PDL
Пример: буква F

$$T_S(F) = (vp + (h \times (vp + h)))$$



Пример: буква A



A.C. Shaw "A Formal Picture Description Scheme as a Basis for Picture Processing Systems"

Нейросетевой подход

Нейросетевой подход. Основные этапы распознавания текста на цифровом изображении

1 этап – детектирование – определение на изображении областей, содержащих текст

...



Нейросетевой подход. Основные этапы распознавания текста на цифровом изображении

1 этап – детектирование – определение на изображении областей, содержащих текст

2 этап – сегментация – выделение позиции отдельных символов

...



Нейросетевой подход. Основные этапы распознавания текста на цифровом изображении

1 этап – детектирование – определение на изображении областей, содержащих текст

2 этап – сегментация – выделение позиции отдельных символов

3 этап – распознавание отдельных символов

Опционально:

4 этап – исправление ошибок

С точки зрения разработки каждый этап – отдельный компонент, который может разрабатываться независимо от других.



1ife -> life

Нейросетевой подход. Детектирование текста на цифровом изображении

Как поступают обычно при детектировании?

Собирают набор данных для обучения детектора, состоящий из тысяч (и более) положительных и отрицательных примеров

Но как быть с текстом? Проблема сложнее детектирования других классов объектов с фиксированной или слабо изменчивой формой.

Подход почти не отличается:
положительные примеры – обрезки букв (слов) соответствующие ожидаемому размеру символов

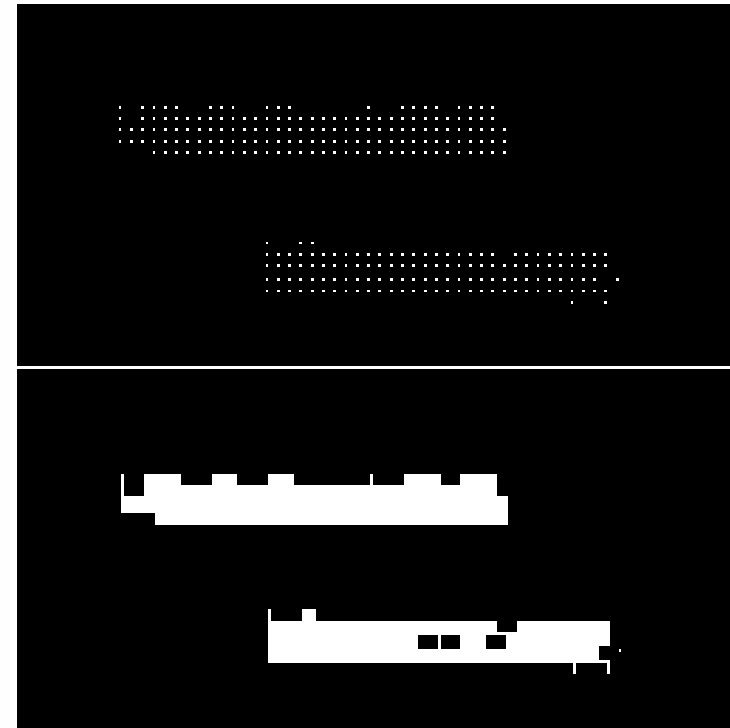
Положительные примеры



Отрицательные примеры

Нейросетевой подход. Детектирование текста на цифровом изображении

Результат применения детектора



После применения детектора выполняется расширение областей так, чтобы слова сливались в одну область

Результат детектирования

Мама мыла раму

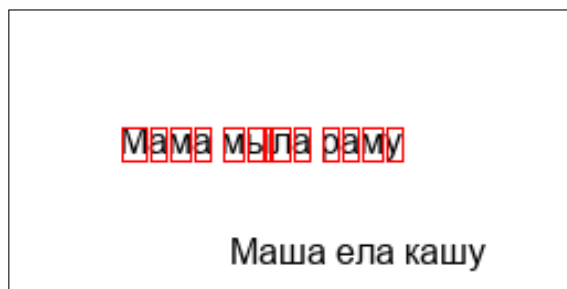
Маша ела кашу

Нейросетевой подход. Обучение сегментатора

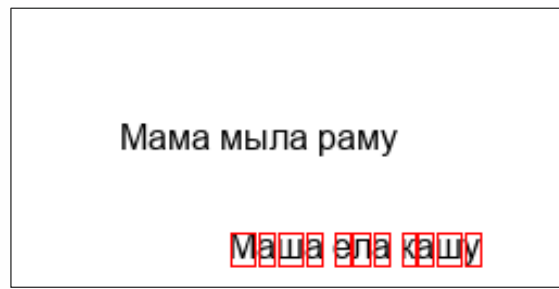
Для обучения сегментатора положительные примеры – разрывы между буквами

Сегментация производится по детектированным областям

Область 1



Область 2



Положительные примеры



Отрицательные примеры



Нейросетевой подход. Обучение классификатора, аугментация

Классификатор обучается различать символы различных классов

Аугментация:

- добавление шума
- геометрические искажения (не только сдвиг, масштаб, поворот)
- наложение фона
- реальные и синтезированные данные





Нейросетевой подход. Современные модели нейронных сетей.

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In Proc. CVPR, pages 2642–2651, 2017.

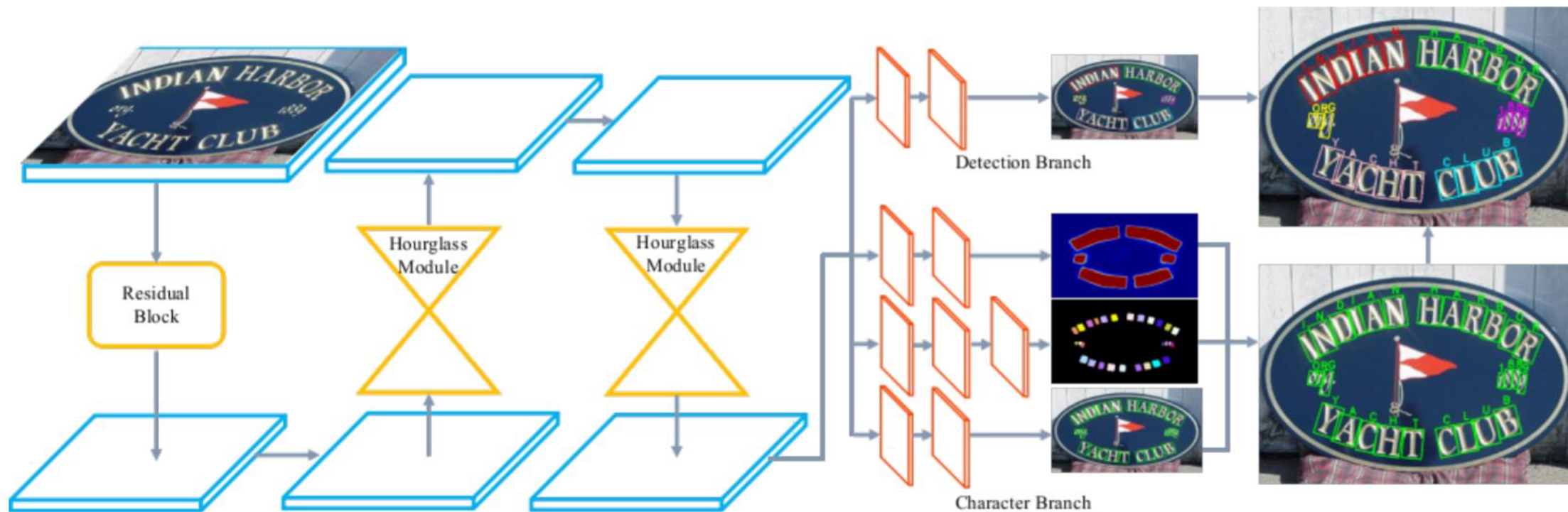
T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun. An end-to-end textspotter with explicit alignment and attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5020–5029, 2018.

X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5676–5685, 2018

P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. 2018.

L. Xing, Zh. Tian, W. Huang, M. R. Scott Convolutional Character Networks. arXiv 1910.07954, 2019

Нейросетевой подход. Современные модели нейронных сетей.



Архитектура Charnet: L. Xing, Zh. Tian, W. Huang, M. R. Scott Convolutional Character Networks. arXiv 1910.07954, 2019

Наборы данных



Наборы данных.

1. **ICDAR 2003** Robust Reading Competitions dataset
 2. **KAIST** Scene Text Database
 3. **SVT** Street View Text dataset
 4. **CBDAR 2011** (IUPR Dataset of Camera-Captured Document Images)
 5. **DIQA** (Dataset for Quality Assessment of Camera Captured Document Images)
 6. **MIDV-500** (Dataset for Identity Documents Analysis and Recognition on Mobile Devices in Video Stream)
- ...

Наборы данных. ICDAR 2003

ICDAR 2003 Robust Reading Competitions
[http://www.iaprtc11.org/mediawiki/index.php/](http://www.iaprtc11.org/mediawiki/index.php/ICDAR_2003_Robust_Reading_Competitions)
ICDAR 2003 Robust Reading Competitions



Наборы данных. KAIST

KAIST Scene Text Database

[http://www.iaprtc11.org/
mediawiki/index.php?title=
KAIST Scene Text Database](http://www.iaprtc11.org/mediawiki/index.php?title=KAIST_Scene_Text_Database)



Наборы данных. SVT

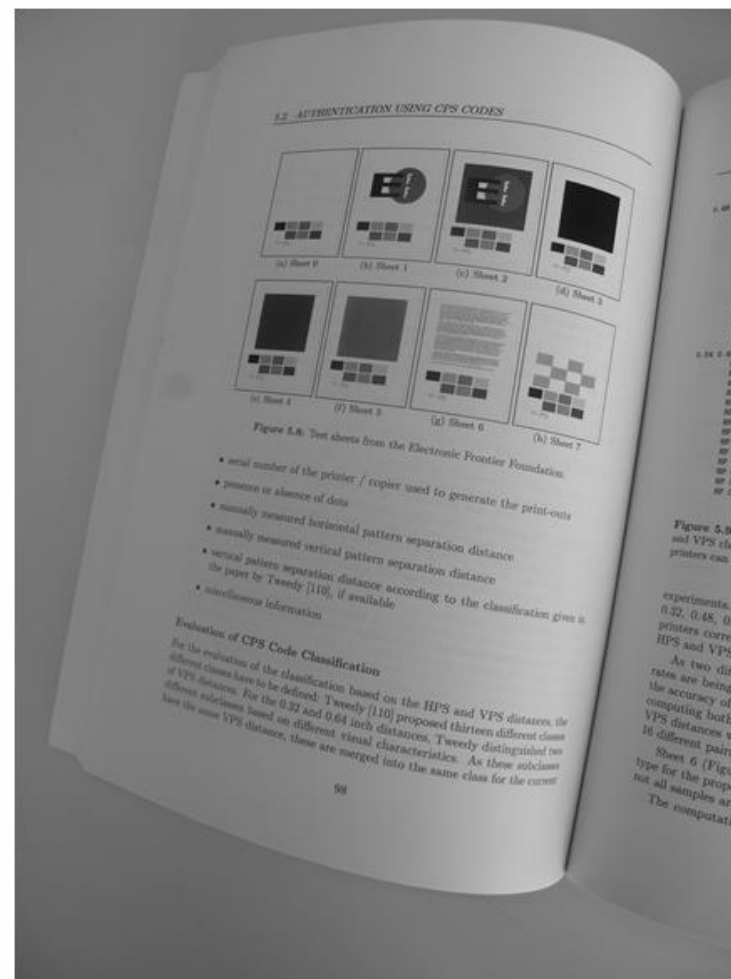
SVT (Street View Text)

http://www.iaprtc11.org/mediawiki/index.php?title=The_Street_View_Text_Dataset



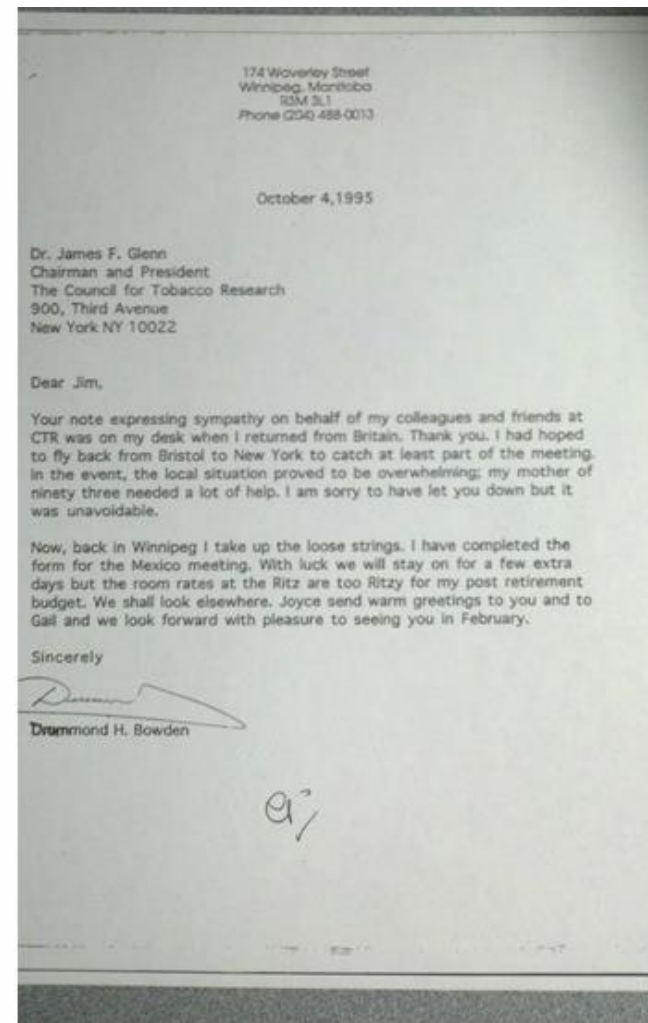
Наборы данных. CBDAR 2011

CBDAR 2011 - различного рода
технические и нетехнические книги с
различными типами макетов.
Искажения, перспективные искажения,
разное разрешение



Наборы данных. DIQA

DIQA - снятые камерой изображения документов, содержащие различные уровни размытия фокуса



Наборы данных. MIDV-500

MIDV-500 - документы, удостоверяющие личность, Видеопоток снят на Samsung Galaxy S3 и Apple iPhone 5.

Размытие, расфокусировка, блики, национальные символы



Библиотеки для ОСR



Библиотеки OCR. Tesseract

Установка под Windows:

<https://github.com/UB-Mannheim/tesseract/wiki>

Под Linux, MacOS,...:

<https://tesseract-ocr.github.io/tessdoc/Home.html>

Интерфейс командной строки:

```
tesseract imagename outputbase [-l lang] [--oem ocrenginemode]  
[--psm pagesegmode] [configfiles...]
```

Для использования с Python:

pytesseract

<https://pypi.org/project/pytesseract/>

Установка:

```
pip install pytesseract  
conda install -c conda-forge pytesseract
```

...



Библиотеки OCR. pytesseract

Импорт библиотеки:

```
import pytesseract
```

Указание пути к исполняемым файлам:

```
pytesseract.pytesseract.tesseract_cmd = r'<full_path_to_your_tesseract_executable>'
```

Распознавание текста на изображении (изображение PIL.Image):

```
pytesseract.image_to_string(<изображение>)
```

Распознавание текста в файле с изображением (или в файлах, перечисленных в текстовом файле):

```
pytesseract.image_to_string(<имя_файла>)
```

Параметры: язык (**lang**), время выполнения (**timeout**)

Получение:

- box-ов текста: `pytesseract.image_to_boxes(...)`
- bbox-ов, номера страниц, строк, коэффициенты достоверности: `pytesseract.image_to_data(...)`
- ориентации и текста: `pytesseract.image_to_osd(...)`



Библиотеки OCR. EasyOCR

Установка:

```
pip install easyocr
```

Импорт библиотеки:

```
import easyocr
```

Создание объекта:

(загрузка модели с указанием языков, Русский – 'ru'):

```
reader = easyocr.Reader(['en'])
```

Распознавание (файл, opencv image):

```
result = reader.readtext('image.jpg', gpu = False)
```

Результат:

```
[ ([x0,y1], [x1,y1], [x2,y2], [x3,y3]), 'Текст', Confidence),  
...]
```



Библиотеки OCR для работы на мобильных устройствах

- Google ML Kit / Firebase
<https://developers.google.com/ml-kit>
- ABBYY Mobile Capture:
<https://www.abbyy.com/mobile-capture-sdk/>
- Ответвления Tesseract Tools for Android
<https://github.com/rmtheis/tess-two>
(на Tesseract 3.05)
- <https://github.com/alexcohn/tess-two>
(на Tesseract 4.1)

Google vision Text API

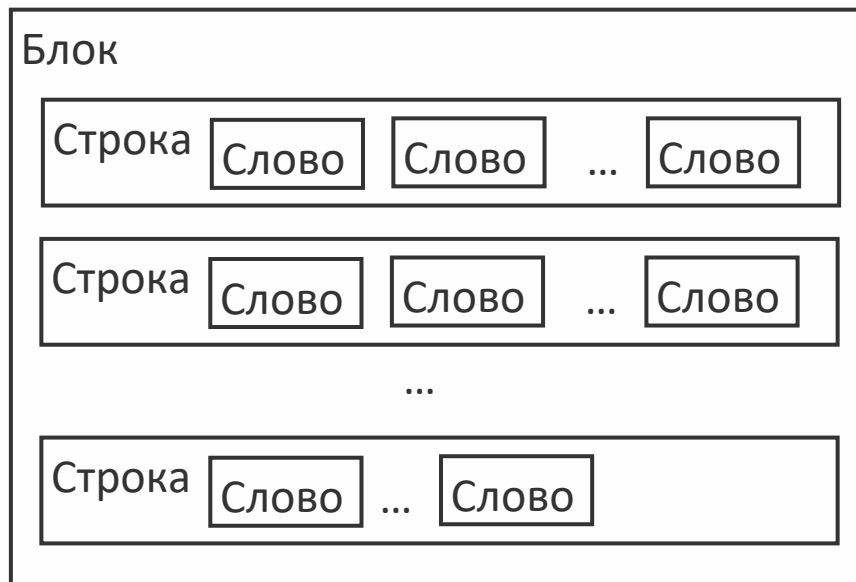
Распознавание текста на нескольких языках (английский, немецкий, французкий,..., но Русского НЕТ)

Основные понятия, связанные со структурой текста:

Блок - набор текстовых строк (абзац, столбец)

Строка - набор слов, расположенных вдоль одной линии

Слово - непрерывный набор буквенно-цифровых символов



Google vision Text API: Подготовка

Подготовительные шаги:

1. Включить репозиторий Google Maven в разделы buildscript и allprojects файла build.gradle на уровне проекта.
2. Добавить зависимости для библиотек ML Kit в файл app / build.gradle (на уровне приложения)
3. Сконфигурировать приложение для автоматической загрузки модели на устройство после инсталляции (необязательно)

Добавить в app/build.gradle:

```
dependencies {  
    // ...  
    implementation 'com.google.android.gms:play-services-mlkit-text-recognition:16.1.1'  
}
```

Добавить в AndroidManifest.xml:

```
<application ...>  
    ...  
    <meta-data  
        android:name="com.google.mlkit.vision.DEPENDENCIES"  
        android:value="ocr" />  
    <!-- To use multiple models:  
    android:value="ocr,model2,model3" -->  
</application>
```




Google vision Text API: Распознавание

1. Создание экземпляра TextRecognizer:

```
TextRecognizer recognizer = TextRecognition.getClient();
```

2. Получение изображения класса InputImage из файла, битмапа, буфера, массива или media.Image, например:

```
InputImage image = InputImage.fromFilePath(context, uri);
```

3. Запуск обработки

```
Task<Text> result = recognizer.process(image)
    .addOnSuccessListener(new OnSuccessListener<Text>() {
        @Override
        public void onSuccess(Text visionText) { /*...*/ }
    })
    .addOnFailureListener( new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) { /*...*/ }
    });
```

4. Обработка результата...



Google vision Text API: Распознавание

Результат хранится в объекте Text.

Обход структуры текста:

```
for (Text.TextBlock block : result.getTextBlocks()) {  
    for (Text.Line line : block.getLines()) {  
        for (Text.Element element : line.getElements()) {  
            // ...  
        }  
    }  
}
```

Для любого из уровней иерархии
доступно получение текста, bbox-a,
узловых точек:

```
String elementText = element.getText();  
Rect elementFrame = element.getBoundingBox();  
Point[] elementCornerPoints = element.getCornerPoints();
```

Примеры распознавания текста в Android Studio

ТОЧНОСТЬ БИБЛИОТЕК OCR

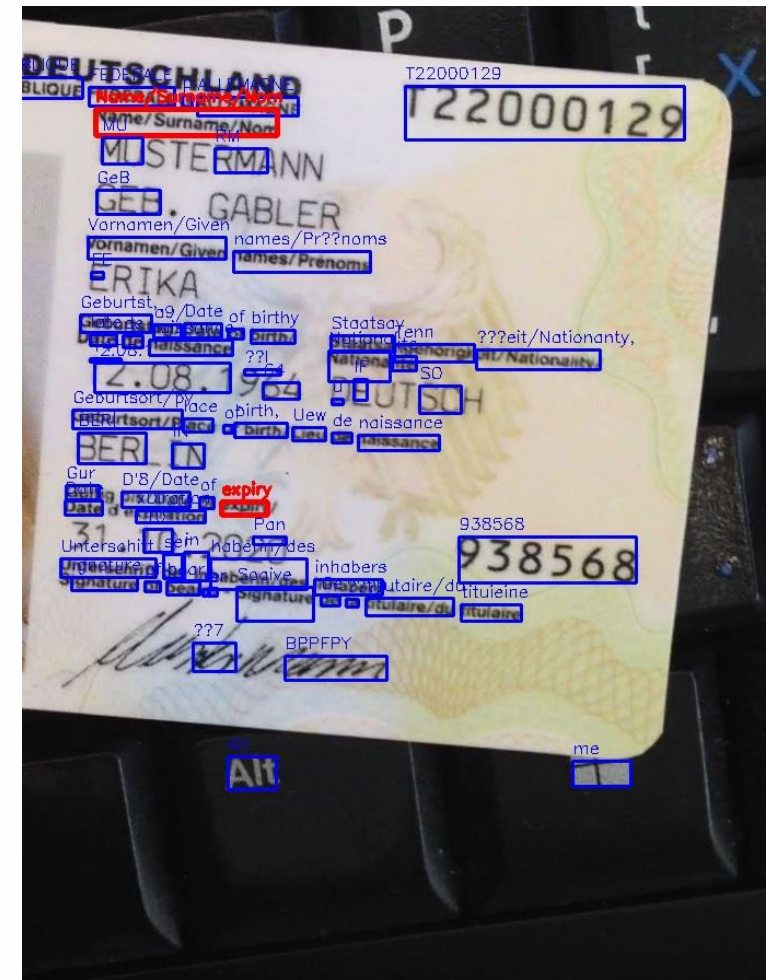
Библиотека OCR	Набор данных					
	ICDAR	KAIST	SVT	CBDAR	DIQA	MIDV-500
Google vision API	61.7	27.2	47.4	67.5	69.9	39.4
Tesseract Tools 3.05 for Android	24.5	8.6	4.0	34.5	54.5	8.4
Tesseract 4.00, psm=3	22.4	10.7	7.9	43.9	76.3	12.4
Tesseract 4.00 psm=11	30.8	17.0	16.4	53.1	75.4	17.3

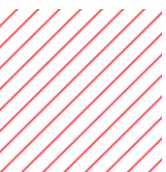
ВРЕМЯ РАБОТЫ БИБЛИОТЕК OCR ДЛЯ МОБИЛЬНОЙ ПЛАТФОРМЫ (НА ОДНО ИЗОБРАЖЕНИЕ, В СЕКУНДАХ)

Библиотека OCR	Набор данных					
	ICDAR	KAIST	SVT	CBDAR	DIQA	MIDV-500
Google vision API	0.156	0.195	0.183	2.436	0.974	0.335
Tesseract Tools 3.05 for Android	0.211	0.359	0.216	13.000	27.110	0.356

Режимы работы Tesseract :

- psm = 3 - полностью автоматическая сегментация страницы, но без ориентации и шрифта (режим по умолчанию).
- psm = 11 - разрезанный текст. Найти как можно больше текста без определенного порядка написания.





Распознавание текста



Спасибо за внимание !